SSB

Local-first database,
cryptographically signed,
and replicated peer-to-peer
Local-first database, cryptographically signed, and replicated peer-to-peer
Local-first database, cryptographically signed, and replicated peer-to-peer
Local-first database, cryptographically signed, and replicated peer-to-peer.
Alice

Log

≥ 500 MB

Bob

Log

Indexes
NGI Pointer
SSB Team

@arj03 @cryptix @staltz @ZELFs

GitHub handles
New SSB Database

+ Log encoded in BIPF
+ Bitvector indexes
+ Prefix indexes
+ JIT index creation
+ Level DB indexes
Old log: JSON

```
{ ... } { ... } 3.3 { ... } 37
```

New log: BIPF "Binary In-place Format"

```
01DA 18 28 65...77 2029 71...6C 28...48
FCF4 4C 44 416C 28...48
```
{"age": 20, "other": { "name": "Alice" } }
DD 01
18
61 67 65 → age
22
14 00 00 00 → 20
28
6F 74 68 65 72 → other
5D
20
6E 61 6D 65 → name
28
41 6C 69 63 65 → Alice
DD 01 → type = 5 (object), length = 2
18 → type = 0 (string), length = 3
61 67 65 → age
22 → type = 2 (integer), length = 4
14 00 00 00 → 20
28 → type = 0 (string), length = 5
6F 74 68 65 72 → other
5D → type = 5 (object), length = 11
20 → type = 0 (string), length = 4
6E 61 6D 65 → name
28 → type = 0 (string), length = 5
41 6C 69 63 65 → Alice
Binary: 0010 0010
Hex: 22
Length Type

Binary: 0010 0010

Hex: 22
4 bytes = Length Type = 2 Integer

Binary: 0010 0010

Hex: 22
4 bytes = Length Type

Binary: 0010 0010

Hex: 22

0  String
1  Buffer
2  Integer
3  Double
4  Array
5  Object
6  Bool / null
7  Reserved
DD 01 → type = 5 (object), length = 27
18 → type = 0 (string), length = 3
61 67 65 → age
22 → type = 2 (integer), length = 4
14 00 00 00 → 20
28 → type = 0 (string), length = 5
6F 74 68 65 72 → other
5D → type = 5 (object), length = 11
20 → type = 0 (string), length = 4
6E 61 6D 65 → name
28 → type = 0 (string), length = 5
41 6C 69 63 65 → Alice
Encoding: JSON approx. 5x faster than BLPF

Decoding: BLPF approx. 6x faster than JSON
New SSB Database

+ Log encoded in B1PF
+ Bitvector indexes
+ Prefix indexes
+ JIT index creation
+ LevelDB indexes
Log in BIPF

Is the message's type a "post"?
Log in BIPF

0 0 1 0 1 0 1 0 1 0 0 1 0

Is the message's type a "post"?
Log in BIPF

0 0 1 0 1 0 1 0 0 0 1 0

Bitvector index

Is the message's type a "post"?
Bitvector index: msg type is post

\[011000100011100\]

\[\downarrow \downarrow \text{AND} \downarrow \downarrow \downarrow\]

Bitvector index: msg was authored by me

\[111000000111000\]

\[\downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow\]

\[01100000000010000\]
Result bitvector

\[01100000001000\]
Result bitvector

0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0

1st 2nd 3rd 4th 5th ...
Result bitvector

\[ 01100000001000 \]

1st 2nd 3rd 4th 5th ...

**Seq-to-offset index**

\[ 0 93 134 157 189 230 271 334 386 417 450 499 560 611 \]
Result bit vector

```
0 1 1 0 0 0 0 0 0 0 0 1 0 0 0
```

1st 2nd 3rd 4th 5th ...

Seq-to-offset index

```
0 93 134 157 189 230 271 334 386 417 450 499 560 611
```

Log
Result bit vector

0 1 1 0 0 0 0 0 0 0 0 1 0 0 0

1st 2nd 3rd 4th 5th ...

Seq-to-offset index

0  93  134  157  189  230  271  334  386  417  450  499  560  611

Log

✓93 ✓134 ✓450
New SSB Database

+ Log encoded in B1PF
+ Bitvector indexes
+ Prefix indexes
+ JIT index creation
+ LevelDB indexes
Log in BIPF

Does the message make a reference to the message with ID "%M3c2DrJsK3..."?
Log in BIPF

In this message, what is the prefix of the msg ID it is referring to?
Log in BIPF

......

0 0 %F189 0 0 %M3c2 0 %M3c2 0 0 0

In this message, what is the prefix of the msg ID it is referring to?
Log in BIPF

Prefix index

In this message, what is the prefix of the msg ID it is referring to?
Prefix index

Does the message make a reference to the message with ID "%.M3c2DrJk3..."?
Prefix index

Does the message make a reference to the message with ID "%.M3c2.DrJk3..."?
New SSB Database

+ Log encoded in BIPF
+ Bitvector indexes
+ Prefix indexes
+ JIT index creation
+ Level DB indexes
API

ssb.db.query(
and(
    type('post'),
    mentions(alice.id)
),
toCallback((err, msgs) => {
    // ...
});
}
API

ssb.db.query(
    and(
        type('post'),
        mentions(alice.id)
    ),
    toCallback((err, msgs) => {
        // ...
    })
)
Performance benchmarks

github.com/ssb-ngi-pointer/db-benchmarks

Roughly 9x faster
than the previous DB

Untested in production
Why make your own DB?
Why make your own DB?

SQLite: table-centric, structured and mutable

SSB: log-centric, unstructured and immutable
Why make your own DB?

SQLite: table-centric, structured and mutable

SSB: log-centric, unstructured and immutable

Databases → B-tree indexes → Bitvector indexes
Local-first database, cryptographically signed, and replicated peer-to-peer.